



Wall Sensing for an Autonomous Robot With a Three-Dimensional Time-of-Flight (3-D TOF) Camera

by Gary Haas and Philip R. Osteen

ARL-TN-423

February 2011

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5066

ARL-TN-423**February 2011**

Wall Sensing for an Autonomous Robot With a Three-Dimensional Time-of-Flight (3-D TOF) Camera

**Gary Haas and Philip R. Osteen
Vehicle Technology Directorate, ARL**

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) February 2011		2. REPORT TYPE Final		3. DATES COVERED (From - To) January 2010–October 2010	
4. TITLE AND SUBTITLE Wall Sensing for an Autonomous Robot With a Three-Dimensional Time-of-Flight (3-D TOF) Camera				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Gary Haas and Philip R. Osteen				5d. PROJECT NUMBER 0BUS01	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-VTA Aberdeen Proving Ground, MD 21005-5066				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TN-423	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This report describes recent work in the field of robotic perception research. The objectives of this research are to (1) program a computer to extract geometric planes characteristic of architectural walls in data from a three-dimensional time-of-flight camera, (2) track the planes through a stream of such data as the camera moved slowly through a plane-rich environment, and (3) serve plane descriptors to processes that desire such service. The programs were implemented at a high level using tools from the open-source Robot Operating System (ROS). Tests performed in simple and complex indoor environments demonstrated the ability to identify and track planes. Postprocessing was performed using a separate process within the ROS environment that receives updated planar information. Preliminary results are presented for dynamic scenes, though sensor motion estimates are necessary to properly quantify results.					
15. SUBJECT TERMS robot, perception, 3-D camera					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 30	19a. NAME OF RESPONSIBLE PERSON Gary Haas
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-278-8867

Contents

List of Figures	iv
List of Tables	v
1. Introduction	1
2. Background	2
2.1 Useful Sensors	2
2.2 SwissRanger	3
2.3 Plane Detection.....	4
3. Method	4
3.1 Planar Extraction	5
3.2 Visualization.....	8
3.3 Labeling and Persisting Planes.....	8
4. Experimentation	10
4.1 Sensor Noise Estimation	11
4.2 Simple Scene	11
4.3 Complex Scene.....	13
4.4 Dynamic Scene.....	16
4.5 Discussion	16
5. Conclusions	18
6. References	20
Distribution List	22

List of Figures

Figure 1. The point-to-plane distance D_p between a point \mathbf{P} and a plane is the projection of the vector \mathbf{w} onto the plane's normal vector \mathbf{n}	5
Figure 2. Simple RANSAC algorithm compared with using local normals.....	6
Figure 3. Principal component analysis used to estimate local normals.....	7
Figure 4. Extended Gaussian images of the same scene, with different thresholds set for number of nearest neighbors (k) to use for local normal estimation.	9
Figure 5. SDs for different distances of an SR-3000 and a vertical plane.....	11
Figure 6. Results from a complex scene.	14
Figure 7. Results from a complex dynamic scene as well as planar tracking plots.	17

List of Tables

Table 1. Excerpts from SwissRanger SR-3000 spec sheet.	3
Table 2. Current definition of a plane.	9
Table 3. Results from experiment 2; the evaluation of RANSAC consistency for a single frame of the simple scene shown in figure 2.	12
Table 4. Results from experiment 3; a stream of data from the simple static scene shown in figure 2.	13
Table 5. Results from experiment 4; the evaluation of RANSAC consistency for a single frame of the complex scene shown in figure 6.	15
Table 6. Results of experiment 5 from a stream of data from the complex scene shown in figure 6.	15

INTENTIONALLY LEFT BLANK.

1. Introduction

Urban warfare is one of the deadliest activities our infantry Soldiers must undertake. Building clearing and street fighting in built-up areas expose our troops to enemy combatants at close range and with little opportunity to assess the situation before the encounter. In this environment, Soldiers routinely use remotely controlled or teleoperated robots to extend situational awareness from nearby defensive positions. This reduces, but does not eliminate, the risk to the Soldier. Fully autonomous robots could be sent ahead of troops to explore environments with minimal supervision. Mapping the environment, particularly the architectural layout and the presence of people, is important information to report to Soldiers before they commit to an area.

The robot is a computer-based machine and must process data from its sensors to extract information useful to itself and the Soldier. In the field of robotic perception, the core challenge is to extract meaningful information about an environment based on raw sensor data. Extracting such information is difficult, so the perception scientist must attempt to simplify the general case to achieve desired results. A robot operating in an environment built by and for people can benefit from some assumptions about its environment. One of the most useful assumptions for an indoor or urban setting is that the environment is dominated by planar structures (walls, floor, ceiling), which are frequently orthogonal. Furthermore, the location of these architectural features is very useful information for situational awareness. Fortunately, sensors capable of responding to these structures have recently become available and are relatively inexpensive.

The focus of this effort is to extract planar objects, such as walls, from the raw data to assist in navigating an unknown environment and to incrementally build maps to assist in planning the exploration of the environment. Section 2 will provide a background of similar work with other sensors, a description of the sensor used for these experiments, and a brief background into other research involving plane extraction. The remainder of the report will focus on the constituents of the process of extracting, identifying, and locating walls in the data from such a sensor. Section 3 will discuss the current state of the research, including the algorithms used to extract and label planes, as well as estimating parameters using visualization tools. Section 4 will present the results of the plane-finding algorithm, in both complex and simple static scenes. Preliminary results are presented for a dynamic scene, but efforts to date have been dedicated to building and developing the tools rather than rigorously evaluating performance in dynamic situations. Finally, section 5 will discuss the conclusions and options to proceed with further research.

2. Background

Two essential characteristics of a wall are that it is planar and vertical. In the context of the interior of a building, a wall may be partially occluded, is typically adjacent to other walls, and may be penetrated by doors or windows. Our robot should not only identify that a wall is nearby, but to map the location of the wall. As our robot is mobile, we need frequent sensor updates to add information from the new sensor viewpoint and to confirm information from previous viewpoints.

2.1 Useful Sensors

While a simple camera can provide this information as long as the robot is moving, location is computationally intensive to extract from the essential two-dimensional (2-D) nature of camera data. Stereoscopic cameras help somewhat, but technologies that intrinsically deliver location are available. If the assumptions of a planar, vertical operating environment reliably hold, a single-line horizontal-scanning laser range finder is useful and inexpensive. A device such as the SICK LMS-200 or the Hokuyo URG-04LX, also called a “line scanner,” senses a wall as a straight line in data converted to Cartesian coordinates. These sensors sweep a laser range finder through a 360° rotation, reporting the range to the scene in a linear frame at an angular resolution of fractions of a degree at a rate of up to 30 Hz. There is an entire body of literature devoted to applying this sort of sensor to navigate through hallways and exploring rooms (Durrant-Whyte and Bailey, 2006; Gutmann and Schlegel, 1996).

The reliability of detecting walls is degraded, however, in environments where the floor is not smooth. This is because sensor roll and pitch convert the wall’s straight line data plot to a curve, confusing the tracking of features from one frame to another. Another problem commonly occurs when the elevation of the sensor causes it to cast its single sensing beam on a cluttered region where occlusions dominate (e.g., the forest of chair legs under a conference table), even though planes dominate at another elevation (e.g., above the conference table).

To overcome these limitations, it is common to mount the line scanner on a tilt actuator, sometimes termed a “nodding” scanner (Harrison and Newman, 2008). Data is then collected from a sequence of planes pivoted at the actuator tilt axis. The degree of tilt is measured and factored into the location computation of the three-dimensional (3-D) point corresponding to a range measurement. This technique retains much of the accuracy of the line scanner and can generate a very dense point cloud. The data frame is a composite of a number of line scanner scans, so the frame rate is low, potentially tens of seconds per frame. Consequently, if a frame is collected from a moving robot, the robot motion must also be factored into the computation of a 3-D point. Since robot motion is generally known only approximately, the accuracy of the generated point cloud suffers.

A sensor that has recently become available is the 3-D time-of-flight (TOF) camera, now available from at least two makers (Mesa Imaging AG and PMD Technologies GmbH). Such a camera integrates a modulated light source synchronized with a complementary metal-oxide semiconductor (CMOS) imager made of “intelligent” pixels. Each pixel calculates the phase difference between the light incident on it and the illumination modulation. The difference is converted to a time of flight, which in turn is converted to a distance. The measurements of all the pixels, organized as a 2-D array, are termed the “range image.” The camera reports the intensity image, the range image, and an image-structured cloud of Cartesian points with origin at the camera, collectively termed a “frame.” (An element of the frame associated with a single “intelligent pixel” will hereafter be called a “pixel.”) Frames are available at rates up to 30 Hz. The expanded vertical field of view (FOV) of this sensor, typically 30°, offers some ability to detect and measure sensor pitch and roll effects in observations of the environment. It also enables observations across a span of elevations, potentially including both the chair leg forest and the walls above the table, for example. A final useful attribute of this sort of sensor is that frames of data are acquired virtually instantly, so low-frequency sensor motion has little effect on the data collected.

2.2 SwissRanger

The sensor selected for the study is a 3-D TOF camera, model SR-3000, from Mesa Imaging AG, of Zurich, Switzerland. This sensor is commonly called a SwissRanger.* Specifications are shown in table 1. The sensor is small and requires little power, making it well suited to a mobile research robot of modest dimension. The pixel count and FOV together provide a generous resolution at the ranges of which the sensor is capable.

Table 1. Excerpts from SwissRanger SR-3000 spec sheet.

Pixel Array Size	176 × 144			
FOV	47.5° × 39.6°			
Output Data (Per Pixel)	Range; x, y, z coordinates; i (intensity)			
	Operating Range (m)	0.3	1	2
	X-Y Resolution (1 Pixel) (mm)	1.5	5.0	10.0
	Distance Resolution (mm)	2.5	6	13
Nonambiguity Range	7.5 m (for 20-MHz modulation)			
Illumination Power (Optical)	1 W (average power) at 850 nm			

Adapted from the Mesa Imaging AG SR-3000 Data Sheet, rev. January 2008; www.mesa-imaging.ch/pdf/SR3000_Flyer_Jan07.pdf (accessed January 2011).

* SwissRanger is a registered trademark of Mesa Imaging AG.

The SR-3000 sensor suffers some limitations not apparent in the spec sheet. The spec sheet cites a “non-ambiguous range” distance limitation of 7.5 m, but in practice the limit is usually the intensity of the modulated light, which diminishes with distance from the sensor. Our experiments seldom range beyond 5 m. Another limitation is ambient light. The spec sheet cautions that the sensor is for “indoor operation only.” Under indoor conditions, distances to an undraped window are completely unreliable. We have not yet attempted to quantify these observations but note them as needing further characterization.

Other effects have been widely studied. In particular, nonlinearities in the range measurements of 3-D TOF cameras have been documented (Lindner and Kolb, 2007; May et al., 2009), along with procedures for calibrating out the worst of the error. These nonlinearities have been measured as a diminishing sinusoid, with amplitude as high as 10 cm with a period as short as 1.5 m. The measurements of interest in fitting planes are typically among nearby neighbors, however, and appear to be highly correlated in the data we have analyzed. Further effort devoted to calibration is likely to improve the accuracy of measurement, but the “factory” calibration has shown itself to be sufficiently accurate for our current application.

2.3 Plane Detection

Two groups, whose writings we have tracked, have published extensively in the realm of extracting planes from point clouds. One group works out of Jacobs University, Bremen, and is anchored by Andreas Birk, Kaustubh Pathak, Jann Poppinga, and Narunas Vaskevicius. This group favors 3-D TOF sensors, prefers a region-growing approach to plane extraction without using local normals (Vaskevicius et al., 2007), and has described an approach to plane matching that supports tracking for simultaneous localization and mapping (SLAM) applications (Pathak et al., 2009).

Another group from Munich Technical University and Willow Garage, including Radu Bogdan Rusu and others (Marton et al., 2009; Rusu, 2009), employ random sampling consensus (RANSAC)-based approaches to extract planes and other geometric primitives from a point cloud data set. The consensus is built on associations between a planar model and features of a given point. While the simplest associations are made using the 3-D location of a point, a more complex method makes use of local point normals as well.

3. Method

Region-based growing methods have been researched previously by the authors with the intent of developing a fast and robust planar region-growing technique. The resulting technique identifies planar patches given a seed point; rather than the traditional region-growing method of identifying all coplanar points to the seed point, the algorithm attempts to quickly find the extents of the coplanar patch. The sacrifice made is that not every point within the extents is

tested, and thus less data is used to define a planar patch. Experiments showed the method to be faster than traditional region growing, though the method suffers the same limitations as any seed-based algorithm. That is, automatic selection of good seed points is very difficult to achieve; though it is a well-studied topic, it has yet to be solved. This issue, combined with the fact that the software platform (ROS.org, 2011a) supports RANSAC-based approaches over region-growing methods, led to the choice to use a RANSAC-based approach to extract planes.

3.1 Planar Extraction

In the simplest case of RANSAC-based plane finding, models are built by fitting the best plane to a random sampling of points from a point cloud data set. The model is then verified according to how many points have a small point-to-plane distance.

As shown in figure 1, a plane may be defined by $ax + by + cz + d = 0$, where \mathbf{n} is the unit normal vector defined by (a,b,c) , and d is the normalized distance to the origin of the reference frame (in our case, the origin of the sensor frame). To find the distance of the plane to a point \mathbf{P} , the point-to-plane distance D_p is derived as follows. The vector \mathbf{v} is the vector between the origin and \mathbf{P} , and the vector \mathbf{d} is the vector from the origin to the closest point on the plane, defined by $\mathbf{d} = d\mathbf{n}$. The vector \mathbf{w} is defined as $\mathbf{w} = \mathbf{v} - \mathbf{d}$. The point-to-plane distance is the projection of \mathbf{w} onto the unit normal \mathbf{n} , given as $D_p = \mathbf{w} \cdot \mathbf{n} = \mathbf{v} \cdot \mathbf{n} - d$.

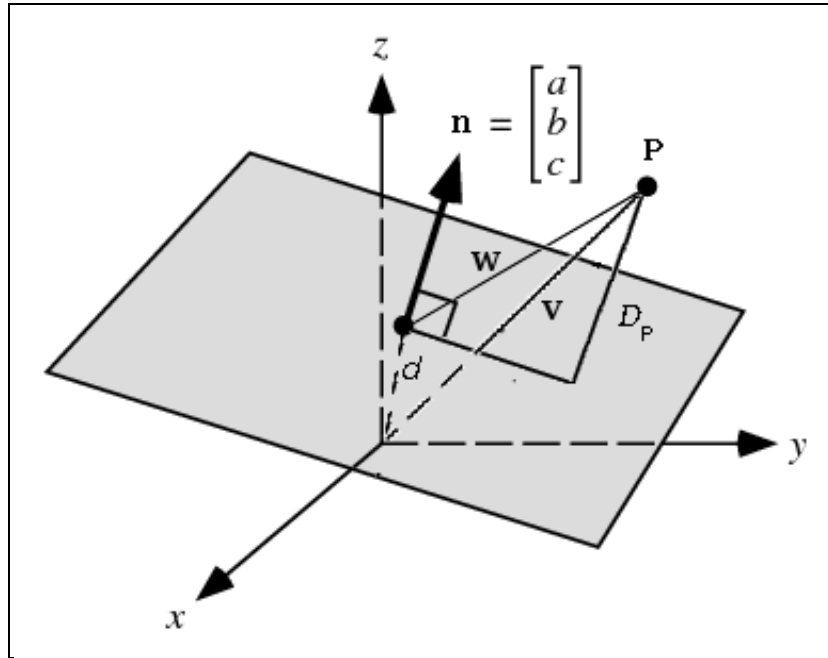


Figure 1. The point-to-plane distance D_p between a point \mathbf{P} and a plane is the projection of the vector \mathbf{w} onto the plane's normal vector \mathbf{n} .

RANSAC uses the point-to-plane threshold to identify all points that belong to the same plane, and the plane with the largest number of points is returned. The inliers to the plane are removed from the original point cloud, and the process is repeated on the outliers of the plane until at least two-thirds of the data set has been segmented. This method requires no preprocessing and is therefore fast when compared with methods that require normal vector estimates at each point. Figure 2 illustrates a shortcoming of the method. The largest planes have been properly identified, but many points are improperly grouped. This is due to the fact that they lie on the mathematical plane defined in the RANSAC algorithm, but they physically belong to a distinct nonparallel plane. Depending on which planar model is randomly created first, the points might belong to one of multiple planes.

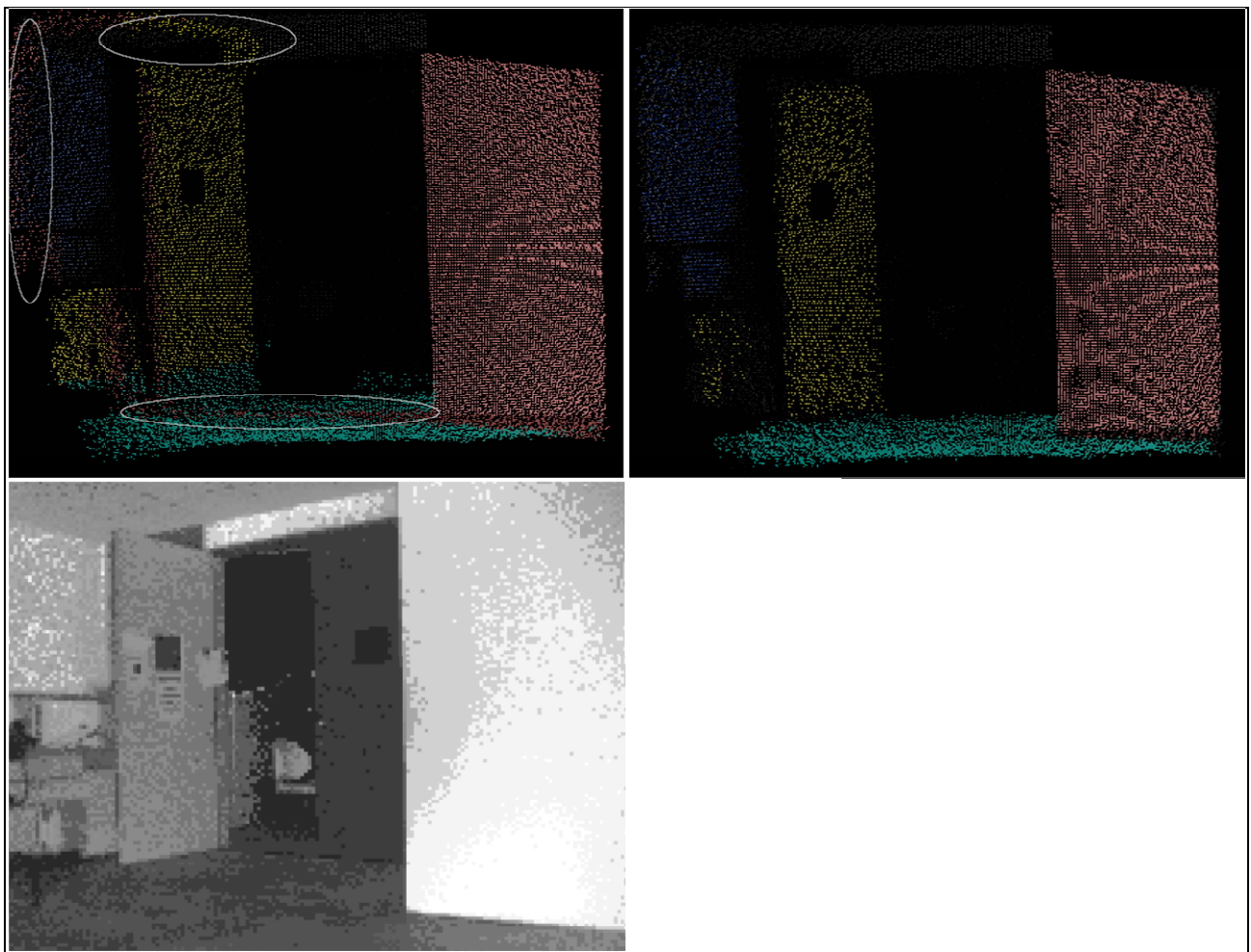


Figure 2. Simple RANSAC algorithm compared with using local normals.

Note: Top left: red and yellow planes contain points that should belong to distinct intersecting planes. Top right: local normals are used to correct these artifacts. Bottom: normalized range image of the scene.

Though possible postprocessing steps could filter these artifacts out, a more robust method is to use local normal estimations at each point for the RANSAC method. The algorithm works as follows. At each point in a point cloud, principal component analysis is used to estimate the point's normal vector. For a query point P_q in a point cloud C , a $(k+1) \times 3$ matrix A , composed of the 3-D coordinates of P_q and its k nearest neighbors, is created as shown in equation 1. Eigenvalue decomposition of this matrix yields the three principal components of the neighborhood around P_q (equation 2). Since, for a pure plane, the variability is entirely in two directions, the eigenvector associated with the eigenvalue with the smallest absolute value λ_3 is also the direction of the plane's normal vector. For relatively small noise in the sensor compared to the planar size, this association can be extended to estimate the normal direction of the plane passing through the neighborhood of points ($\mathbf{n} \approx \mathbf{v}_3$). This normal direction constitutes the estimated normal direction for P_q . The normal vector \mathbf{n} and the two major eigenvectors defining the plane in the neighborhood of the query point are depicted in figure 3.

$$A = \begin{bmatrix} x_q & y_q & z_q \\ x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_k & y_k & z_k \end{bmatrix}. \quad (1)$$

$$(A - \lambda_i I)v = 0 \Rightarrow \{(v_1, \lambda_1), (v_2, \lambda_2), (v_3, \lambda_3)\}. \quad (2)$$

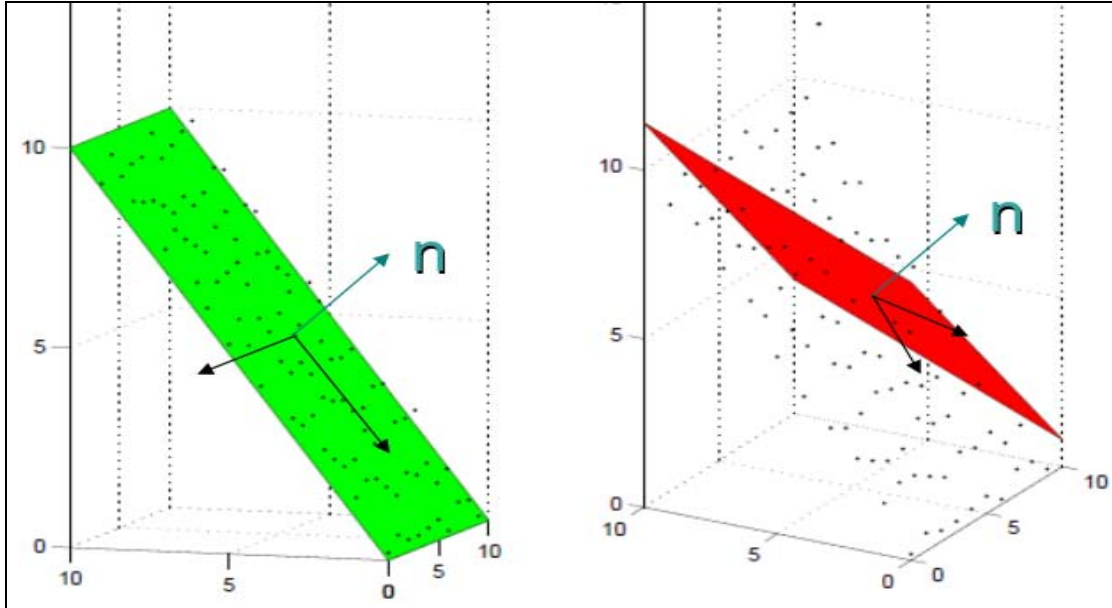


Figure 3. Principal component analysis used to estimate local normals.

Note: At left, the neighborhood contains coplanar points and a good local normal \mathbf{n} is estimated. At right, the presence of a neighbor point with a large offset affects normal estimates.

This local normal provides an extra feature to be used for planar identification. Now, not only must a point be within a certain projected distance of a planar model, but its local normal must also be approximately parallel to the model normal. This method is shown to yield improved results (also shown in figure 2), though the local normal estimation does increase the complexity of the algorithm.

Both the projected distance threshold and the normal angle distance threshold are input to an ROS custom method, derived from the *SACSegmentationFromNormals* class (ROS.org, 2011b), for performing local normal-based RANSAC. The current default values of these parameters are 7.5 cm and 0.05, respectively. These values were determined by qualitative evaluation using visualization tools.

3.2 Visualization

The local normal information is useful for both plane identification and visualization purposes. Visualization is useful for, among other things, approximating a value for the number of neighbors k to use for local normal calculation. Local normal vectors are associated with each point in a point cloud, therefore unit vectors can be added to points directly. A more efficient alternative is to form an Extended Gaussian Image (EGI) (Horn, 1984).

The EGI, for our purposes, is simply a unit sphere that illustrates the distribution of local normal directions in a point cloud. The set of all local normal vectors in a point cloud is oriented such that the tails of the unit vectors are all connected at a point. This point will be the center of a unit sphere, with the unit normals intersecting the unit sphere at a point on the sphere. This point provides an intuitive visualization of the direction of a normal, and this representation over the entire point cloud is the EGI.

Examples of the usefulness of the EGI are shown in figure 4. The same algorithm is run with two different values of k . In a planar region, the EGI should have tight clusters of points. For the given planar region, k is shown to have a strong impact on the EGI. Since the accurate unit normal estimation is a requirement for RANSAC, successful plane finding depends on a well-chosen k parameter.

3.3 Labeling and Persisting Planes

We currently identify the primary planes in a given frame of point cloud data, then attempt to associate planes in the current frame with planes from past frames. Associations between two planes are made in an analogous way to the RANSAC method already described. The two parameters required for normal-based RANSAC are the point-to-plane distance threshold and the normal angle difference threshold. The same criteria are used to combine plane i from the list of current planes in the current frame, returned by the RANSAC algorithm, with plane j from the list of previously identified planes. Since the algorithm depends on point-to-plane distances, the centroid of plane i is projected onto plane j . Further, the plane normal is compared with those

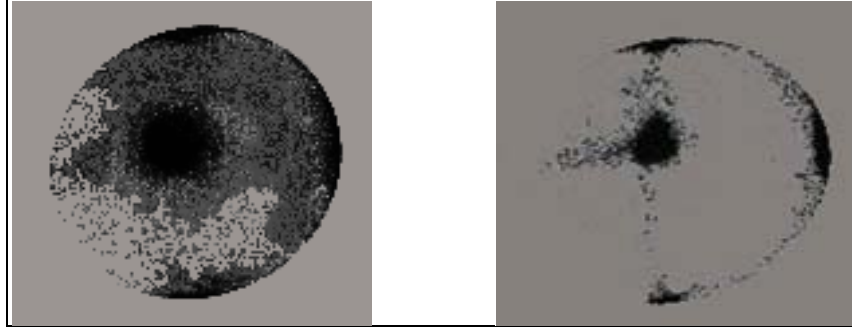


Figure 4. Extended Gaussian images of the same scene, with different thresholds set for number of nearest neighbors (k) to use for local normal estimation.

Note: At left, $k = 20$ yields poor results, with sensor measurement noise leading to noisy normal estimates. At right, $k = 80$ mitigates the effects of sensor errors by increasing the neighborhood sample size.

identified in previous frames. If no match is found, it is added to a dynamically growing list of planes. The planar definitions, shown in table 2, are much more compact than the point data they represent. The list of planes is broadcast within the ROS environment to any interested node, which may wish to use the planar definitions for higher-level processing. If a plane were defined in a previous frame but is not currently visible, it still remains in the plane list in case it is seen again in the future.

Table 2. Current definition of a plane.

Planar Data Structure		
Field Type	Field Name	Field Description
bool	is_current	Distinguishes between planes that exist in the current frame and those that were seen previously
int	num_pts	Number of points represented by the plane
int	mse	Mean-squared error of points (error is measured as point-to-plane distances)
float*	coef	Coefficients (a,b,c,d) that define the plane
float*	centr	Centroid of the set of points belonging to the plane
float*	conv_hull	Convex hull of the set of points belonging to the plane (for future work)
uint	decay	Function of time from when plane was last seen (for future work)

Given proper associations between current planes and those of prior frames, a logical progression would be to attempt to find the best rotation and translation between the current and previous planes, thus providing an ego-motion estimate. This is a problem currently being researched by Pathak et al. (2010), although their plane extraction methods differ from ours. There are other methods currently applied to sets of points that might be extended to sets of planes, such as the Iterative Closest Point (ICP) algorithm. Investigation into these topics is left as future work.

4. Experimentation

Experiments were performed to identify the sources of uncertainty of the planar estimation. For the first experiment, the uncertainty induced by sensor noise is quantified. For the remainder of the experiments, we attempted to isolate uncertainty induced by RANSAC alone vs. the uncertainty of the complete process, which incorporates uncertainty from both the sensor data and the RANSAC algorithm.

To isolate the effects of RANSAC, we extract a single frame from the SwissRanger and repeatedly run the RANSAC algorithm on it as if it were a live data stream, thus eliminating sensor noise as a source of uncertainty. The variability of planar estimates therefore results purely from the random nature of the RANSAC algorithm. Then, the full algorithm is run with live sensor data, and the results are analyzed.

Statistical measures of variability for each plane are described as follows. Results using these descriptions are reported in tables 3–6.

- Number of points: The number of points reported as belonging to the plane, both mean and standard deviation (SD), *npts_mean* and *npts_std*. The number of times the plane was represented in the results is also reported as *percent_id*, with a value of 1.0 representing 100%.
- Plane angle: The planar normal, represented in sensor coordinates. We report the mean unit vector *normal_means* as point coordinates in the sensor coordinate frame and circular standard deviation *normal_stds* of plane normals for the plane over all frames.
- Centroid: The mean *centr_mean* and SD *centr_stdXYZ* of the centroid of the points belonging to the plane. These statistics are reported as point coordinates in the sensor frame.
- Centroid projection: The non-negative point-to-plane distance of the plane centroid onto the plane defined by the *normal_means* vector. The mean *centr_proj_mean* and SD *centr_proj_std* over all frames are reported.
- Within-plane point error: The root mean square error (RMSE) measured along the plane normal from a point to its associated plane for an individual plane within a frame. We report the mean *RMS_mean* of the RMSE over all frames and the corresponding SD *RMS_std*.
- Distance to corrected plane: The distance from the sensor center to a plane parallel to the mean plane (*normal_means*) through the frame centroid. We report the mean distance *plane_dist_mean* and SD *plane_dist_std*.

4.1 Sensor Noise Estimation

The objective of experiment 1 was to determine the variability of the sensor itself. The measure of variability is the SD across frames of the distance to a static planar target. The distance selected for the measure was not the measured range, but the component of the range along the axis of the camera (the “z” component of the Cartesian point in sensor coordinates, reported by the 3-D camera). Use of the Cartesian z component allows easy comparison of a pixel to its neighbors in the target frame and is consistent with the intended use as a sensor of planes.

As shown in figure 5, the SD measure varies in a nonlinear fashion with distance from the center pixel. At distances between 0.5 and 5 m, pixels with SD >5 cm occur only at the corners of the image. The majority of pixels have SD between 0.5 and 2 cm. This value is near the resolution of the sensor claimed in the SwissRanger spec sheet and is consistent with the value reported in Kahlmann et al. (2007).

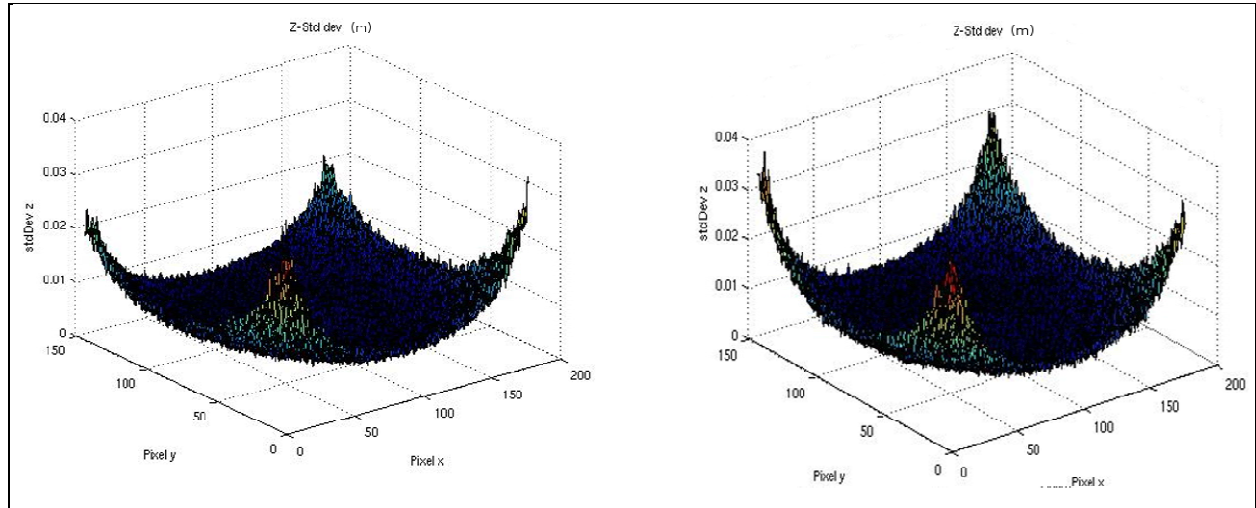


Figure 5. SDs for different distances of an SR-3000 and a vertical plane.

Note: For the left and right plots, the distances to the wall are 1.98 and 2.31 m, respectively. Tests showed that for under 5 m to the wall, only the corners of the image had SDs larger than 5 cm.

4.2 Simple Scene

In experiments 2 and 3, sensor data from a simple scene containing three strong planes (shown in figure 2) was processed by our method of plane extraction. In experiment 2, planes were extracted from a single frame of the simple scene. This was repeated 30 times using the same sensor data. Since the sensor data is exactly the same in each iteration, the variability in results must be a consequence of the RANSAC method of plane extraction. The results from experiment 2 are shown in table 3.

Table 3. Results from experiment 2; the evaluation of RANSAC consistency for a single frame of the simple scene shown in figure 2.

Measures of Variability	Plane_1	Plane_2	Plane_3
npts_mean	8.5877e+03	1.9456e+03	1.1906e+03
npts_std	14.5150	120.3187	101.0541
percent_id	1	1	0.6667
normal_means (m)	0.4854	-0.7549	-0.8054
	7.1067e-05	0.0431	0.0380
	-0.8743	-0.6544	-0.5928
normal_stds (deg)	0.7014	4.0932	4.9233
centr_mean (m)	-0.8943	1.1677	2.1528
	0.1480	0.1796	0.9314
	3.3595	5.3273	5.9246
centr_stdXYZ (m)	0.0016	0.0347	0.0132
	8.1630e-04	0.0450	0.0699
	8.8596e-04	0.0373	0.0194
centr_proj_mean (m)	6.2594e-05	0.0071	0.0046
center_proj_std (m)	5.9724e-05	0.0052	0.0037
RMS_mean (m)	0.0181	0.0360	0.0345
RMS_std (m)	0.0053	0.0046	0.0089
plane_dist_mean (m)	3.3713	4.3600	5.2085
plane_dist_std (m)	8.7293e-05	0.0089	0.0060

In experiment 3, planes were extracted from a stream of 10 frames of the simple static scene. In addition to RANSAC variability, the variability of the sensor affects results. However, the simplicity of the scene makes matching errors unlikely. The results of experiment 3 are shown in table 4.

Table 4. Results from experiment 3; a stream of data from the simple static scene shown in figure 2.

Measures of Variability	Plane_1	Plane_2	Plane_3
npts_mean	8.5732e+03	1.9339e+03	1.2723e+03
npts_std	22.2351	185.4798	53.1890
percent_id	1	1	0.6000
normal_means (m)	0.4889	-0.7433	-0.7906
	1.7207e-04	0.0433	0.0555
	-0.8723	-0.6675	-0.6098
normal_stds (deg)	1.1525	4.4461	3.1046
centr_mean (m)	0.8976	1.1689	2.1455
	0.1454	0.1500	0.9238
	3.3636	5.3328	5.9169
centr_stdXYZ (m)	0.0041	0.0302	0.0045
	0.0015	0.0825	0.0196
	0.0026	0.0428	0.0089
centr_proj_mean (m)	0.0011	0.0085	0.0045
center_proj_std (m)	0.0015	0.0050	0.0023
RMS_mean (m)	0.0189	0.0363	0.0344
RMS_std (m)	0.0080	0.0076	0.0065
plane_dist_mean (m)	3.3730	4.4222	5.2530
plane_dist_std (m)	0.0019	0.0103	0.0055

4.3 Complex Scene

In experiments 4 and 5, sensor data from a complex scene containing a number of planes was processed by our plane extraction method. The complex scene used for testing is shown in figure 6. Figure 6 also shows a top-down orthogonal view, to illustrate the depth offset of the vertical planes and a range image.

In experiment 4, planes were extracted from an archive of 30 frames of the complex scene. This experiment presents the algorithm with a stronger challenge in matching, but because the data is identical from frame to frame, the only error can come from the random nature of the RANSAC algorithm. The results of experiment 4 are shown in table 5.

In experiment 5, planes were extracted from a stream of 10 frames of a complex scene. As in experiment 3, errors from both RANSAC and the sensor affect results. In this experiment, a new measure of variability is added: the number of new (different from any encountered in the previous frame) planes encountered in a frame. The scene is static, so the number of planes should remain constant after initialization. If a plane is found for which there is no existing match, it is recorded as a new but different plane. In a static scene such as this one, this indicates a matching error. A frame-to-frame plot of this measure, as well as the total number of planes discovered in experiment 5, is presented in the fourth quadrant of figure 6. Other results of experiment 5 are shown in table 6.

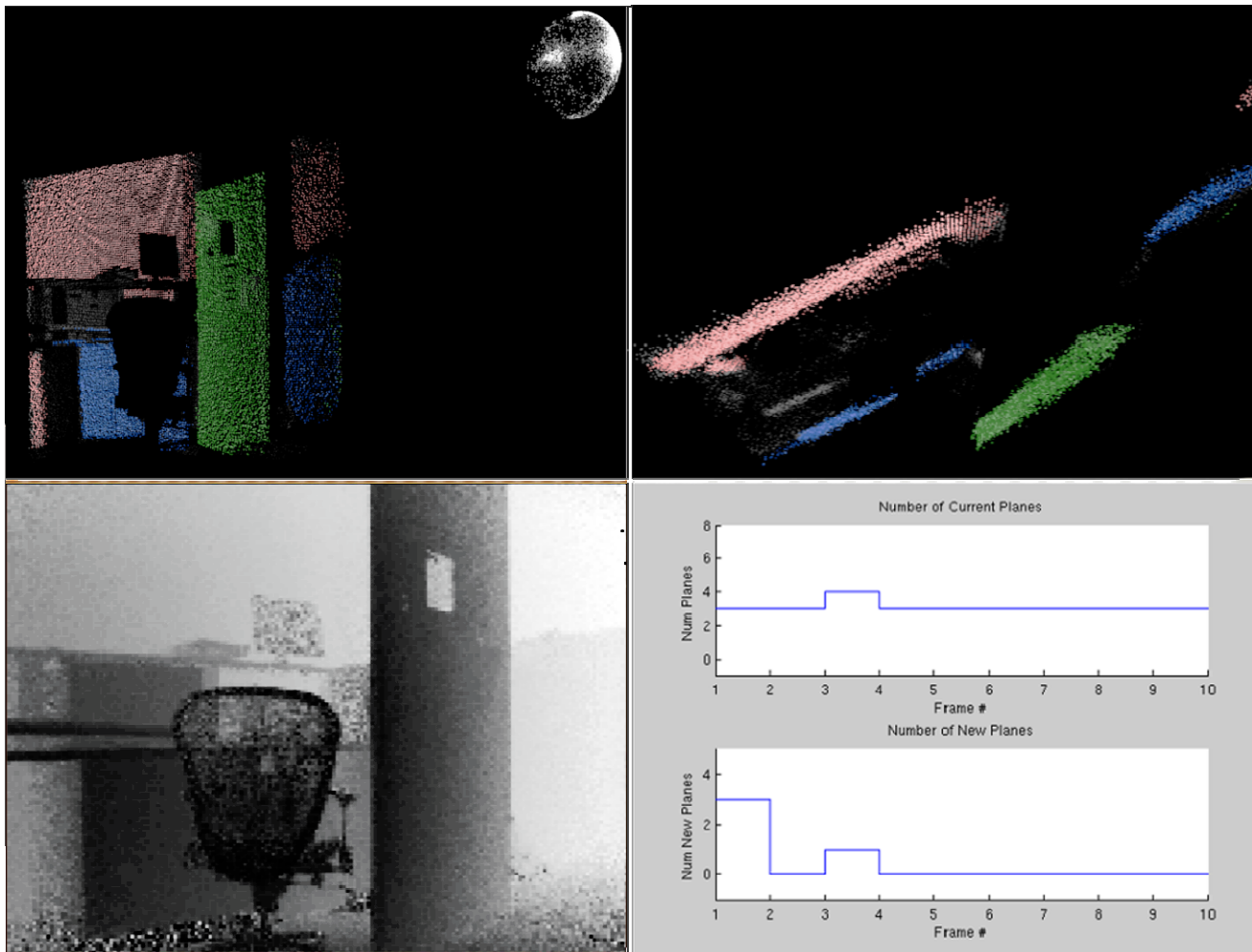


Figure 6. Results from a complex scene.

Note: Top left: predominant planes are displayed along with extended Gaussian image. Top right: same data from top-down view. Bottom left: range image corresponding to this frame. Bottom right: the number of new planes entering the scene is tracked, along with the number of current planes in a given frame. The step in both graphs at frame 3 demonstrates that a new plane was identified in frame 3 but then disappeared in frame 4. This is the result of a tracking error in experiment 5.

Table 5. Results from experiment 4; the evaluation of RANSAC consistency for a single frame of the complex scene shown in figure 6.

Measures of Variability	Plane_1	Plane_2	Plane_3
npts_mean	5.2804e+03	4.7081e+03	2.6738e+03
npts_std	74.1241	82.7101	213.8191
percent_id	1	1	1
normal_means (m)	-0.5936	-0.8128	-0.6961
	0.0415	0.0155	0.0596
	-0.8037	-0.5824	-0.7155
normal_stds (deg)	1.1573	1.5767	2.6868
centr_mean (m)	0.4010	-0.5047	-0.1621
	0.9685	0.0878	-0.3424
	4.1867	3.2650	3.6768
centr_stdXYZ (m)	0.0331	0.0333	0.0755
	0.0197	0.0062	0.0308
	0.0269	0.0538	0.0770
centr_proj_mean (m)	0.0015	0.0033	0.0043
center_proj_std (m)	0.0014	0.0032	0.0054
RMS_mean (m)	0.0289	0.0265	0.0348
RMS_std (m)	0.0065	0.0130	0.0420
plane_dist_mean (m)	3.5625	1.4900	2.5382
plane_dist_std (m)	0.0021	0.0046	0.0069

Table 6. Results of experiment 5 from a stream of data from the complex scene shown in figure 6.

Measures of Variability	Plane_1	Plane_2	Plane_3
npts_mean	5.3143e+03	4.9971e+03	2.8782e+03
npts_std	88.8495	173.4964	188.5623
percent_id	1	1	1
normal_means (m)	-0.5826	-0.8005	-0.6846
	0.0485	0.0160	0.0417
	-0.8113	-0.5991	-0.7277
normal_stds (deg)	1.2900	2.2566	1.8974
centr_mean (m)	0.4051	-0.5689	-0.2208
	0.9210	0.0597	-0.3495
	3.9899	3.1275	3.5251
centr_stdXYZ (m)	0.0274	0.0320	0.0623
	0.0156	0.0162	0.0107
	0.1237	0.1174	0.1020
centr_proj_mean (m)	0.0742	0.0611	0.0700
center_proj_std (m)	0.0476	0.0408	0.0449
RMS_mean (m)	0.0279	0.0273	0.0290
RMS_std (m)	0.0080	0.0134	0.0079
plane_dist_mean (m)	3.4284	1.4174	2.4288
plane_dist_std (m)	0.0915	0.0762	0.0863

4.4 Dynamic Scene

Finally, in experiment 6, planes were extracted from a stream of 10 frames of a complex scene (the interior of an office divided into cubicles) in which the sensor moves slowly, with both rotational and translational components. The sensor has no component reporting its pose, and all sensor measurements are in the sensor reference frame, so when the sensor moves, it perceives that the scene and all its components have moved. This experiment challenges the robustness of the tracking algorithm, since the tracking uses the prior definition of the plane with no a priori motion estimation to associate planes.

Figure 7 shows two consecutive frames from experiment 6 that differ visibly as a consequence of sensor motion. Surfaces that change color from one frame to the next indicate that our primitive tracking algorithm was unable to associate a plane from the earlier frame to a plane in its successor. The “new plane” plot also shows evidence of difficulties in matching planes frame-to-frame. While small values of the “new plane” plot may indicate that a planar object has entered or exited the FOV or that a matching error has occurred, large values of the number of new planes probably indicate that sensor motion between frames was too rapid to be accommodated by the algorithm. One can see from the two colored frames shown that the far wall is consistently tracked while the side wall and ceiling are mismatched.

4.5 Discussion

The degree to which two planes agree may be determined from a comparison of their normal vectors and distances to the origin. Since a plane’s distance to the origin depends on the estimate of the normal, which in turn depends on the location of the points belonging to the plane, we use the decoupled centroid projection measure to assess precision.

For the simple scene, the results from the single data set and the stream of data indicate consistent plane extraction. Differences between the mean normal vectors of each plane in table 3 with associated planes in table 4 are all $<2^\circ$, and differences between the projected centroids are all <3 cm, which is on the order of magnitude of sensor noise.

For the complex scene, the mean normal vectors are also $<2^\circ$ different between the single data set and the data stream cases, but the projected centroid measures differ significantly between the two cases. The single frame results are more consistent, as expected, than those of the data stream. The SD of the centroid projection for the data stream is almost 5 cm for each plane.

A few large outlier points would normally explain such variability; however, this is not likely to be the case since large outliers are filtered by the RANSAC distance threshold. Another possible explanation is that a larger number of small outliers (those belonging to a parallel but offset plane) contribute to the uncertainty. This is consistent with the scene, shown in figure 6, where drawers or walls separated by a door may or may not be classified as belonging to a common plane.

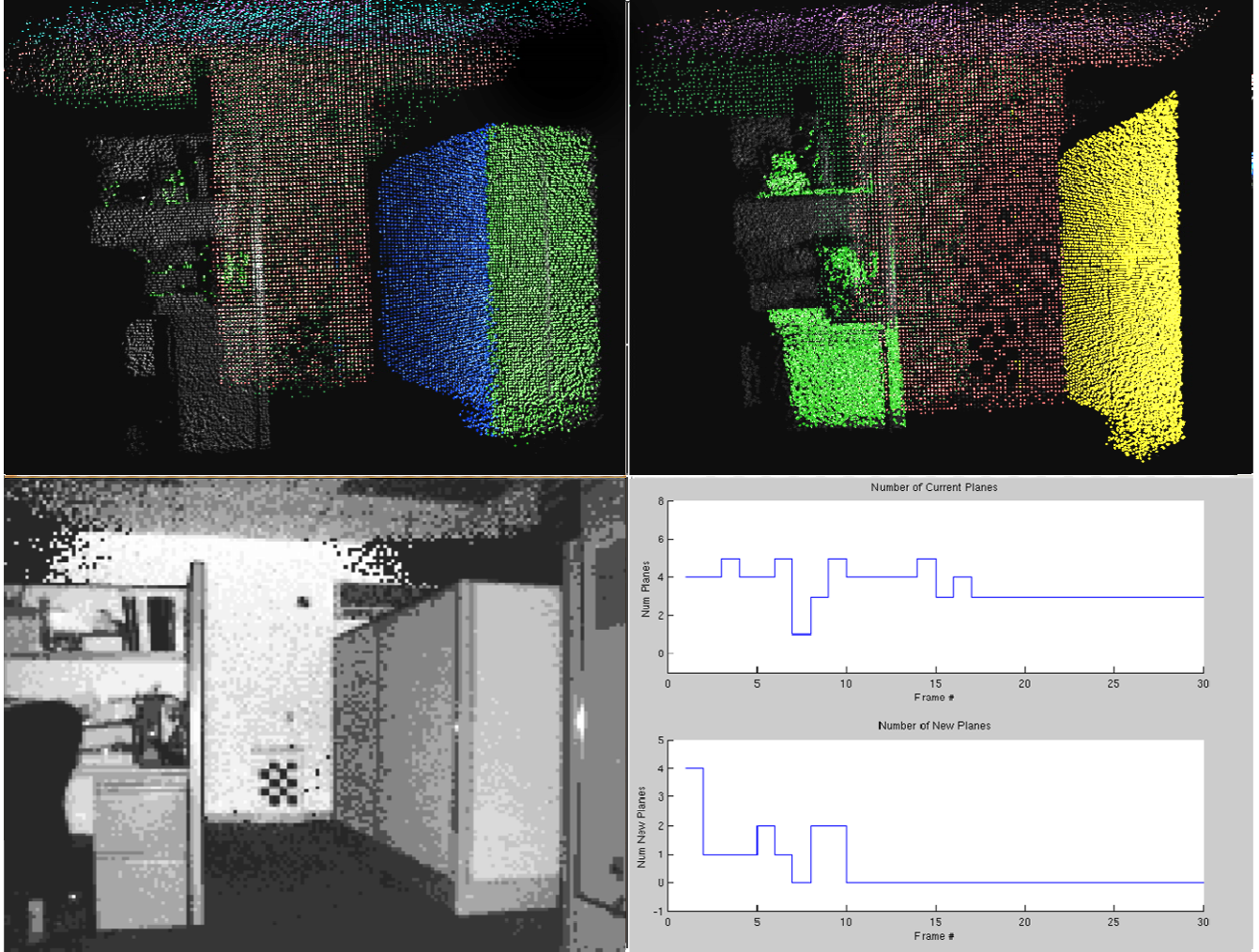


Figure 7. Results from a complex dynamic scene as well as planar tracking plots.

Note: Top left: results for a frame of the scene. Top right: results for a subsequent frame. The side wall is incorrectly added as a new plane (from blue to yellow), while the far wall is tracked properly. Bottom left: range image of the scene. Bottom right: plots of the number of current planes in a scene as well as the number of new planes in the scene. Large discontinuities in the plots are due to high rates of rotation and translation.

Comparisons of descriptive statistics collected on plane parameters (centroid and plane normal) showed no apparent pattern of difference between results from repeated treatments of a single frame and results from treatments of multiple frames from a single static scene (sections 4.2 and 4.3). It is not clear that the measures are adequate to differentiate between noise from RANSAC and noise from the sensor itself. It may be desirable in subsequent work to lump these two sources of variability, as the results of our plane-finding approach appear “good enough” for the intended use.

5. Conclusions

A method for finding predominant planes in a scene using a 3-D TOF camera has been presented. We have demonstrated the ability to find and track planes in a predominantly planar scene, with potential for real-time robotic applications (depending on the parameter configuration, the entire process runs at 2–10 Hz). Much of the code is not optimized, and there are currently costly routines that exist for visualization purposes only.

The effort to date has largely constituted familiarization with the literature, the sensor, and the toolkit selected. The conclusion is that the ROS framework provides useful tools for extracting planes, and that plane extraction using SwissRanger data is sufficient to begin investigating methods to improve performance in dynamic scenes.

The current state of the research presents various options for future work, which require some prioritization. Using the estimates in tracking will require a robust matching algorithm to deal with the variability in the plane estimates.

Also, to properly test the ability of a robot to track planes in a dynamic environment, localization measurements are needed to estimate egomotion. Motion capture systems can be used for precise motion measurement, or onboard sensors could independently provide validation using localization techniques.

Alternatively, localization techniques usually applied to point or pixel features in a scene might be extended to the case of a small number of plane features. This would reduce the computational complexity of an ICP-based approach, and the planar features should be less prone to false registration than a large point cloud. If localization is achievable, then a plane-based SLAM technique could be developed.

As mentioned earlier, automatic seed selection for region growing techniques could be investigated for the purpose of improving plane-finding performance compared with local normal-based RANSAC. Possible approaches could include gradient-based processing on either the point cloud data or the associated range/intensity image to identify relatively nonunique points, which make good seed candidates.

Furthermore, although we are currently concerned with extracting planes from a data set, similar methods are shown by Rusu (2009) to successfully extract other geometric primitives. These geometric primitives are often used to build semantic maps of an environment, which use geometric primitives to abstract higher-level object definitions, such as tables, doors, or ceilings for the planar primitive.

Finally, experiments should be run that quantitatively identify good parameters to use to optimize plane extraction and matching. This will likely require a study of RANSAC parameters. Quality will be assessed in terms of (at least) extraction accuracy and process runtime.

6. References

- Durrant-Whyte, H.; Bailey, T. Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms. *Robotics and Automation Magazine* **2006**, *13*, 99–110.
- Gutmann, J. S.; Schlegel, C. Amos: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments. *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots (Eurobot'96)*, Kaiserslautern, Germany, 1996.
- Harrison, A.; Newman, P. High Quality 3-D Laser Ranging Under General Vehicle Motion. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, 19–23 May 2008; pp 7–12.
- Horn, B. K. P. Extended Gaussian Images. *Proceedings of the IEEE* **1984**, *72* (2), 1671–1686.
- Kahlmann, T.; Remondino, F.; Guillaume, S. Range Imaging Technology: New Developments and Applications for People Identification and Tracking. *Proceedings of SPIE* **2007**, *6491*, C-1–C-12; DOI: 10.1117/12.702512.
- Lindner, M.; Kolb, A. Calibration of the Intensity-Related Distance Error of the PMD TOF-Camera. *Proceedings of SPIE: Intelligent Robots and Computer Vision XXV* **2007**, *6764*, W-1–W-8.
- Marton, Z. C.; Rusu, R. B.; Jain, D.; Klank, U.; Beetz, M. Probabilistic Categorization of Kitchen Objects in Table Settings With a Composite Sensor. *Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, 11–15 October 2009.
- May, S.; Droschel, D.; Holz, D.; Fuchs, S.; Malis, E.; Nüchter, A.; Hertzberg, J. Three-Dimensional Mapping With Time-of-Flight Cameras. *Journal of Field Robotics* **2009**, *26*, 934–965.
- Pathak, K.; Birk, A.; Vaskevicius, N.; Pfingsthorn, M.; Schwertfeger, S.; Poppinga, J. Online 3-D SLAM by Registration of Large Planar Surface Segments and Closed Form Pose-Graph Relaxation. *Journal of Field Robotics* **2010**, *27* (1), 52–84.
- Pathak, K.; Vaskevicius, N.; Poppinga, J.; Pfingsthorn, M.; Schwertfeger, S.; Birk, A. Fast 3-D Mapping by Matching Planes Extracted From Range Sensor Point-Clouds. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, 2009.
- ROS.org, 2011a. <http://www.ros.org> (accessed February 2011).

- ROS.org, pcl::SACSegmentationFromNormals< PointT, PointNT, > Class Template Reference, 2011b. http://www.ros.org/doc/api/pcl/html/classpcl_1_1SACSegmentationFromNormals.html (accessed February 2011).
- Rusu, R. B. Semantic 3-D Object Maps for Everyday Manipulation in Human Living Environments. Ph.D. Thesis, Technische Universität München, Germany, 2009.
- Vaskevicius, N.; Birk, A.; Pathak, K.; Poppinga, J. Fast Detection of Polygons in 3-D Point Clouds From Noise-Prone Range Sensors. *Proceedings of the International Workshop on Safety, Security, and Rescue Robotics*, Rome, 2007.

NO. OF
COPIES ORGANIZATION

1 DEFENSE TECHNICAL
 (PDF INFORMATION CTR
 only) DTIC OCA
 8725 JOHN J KINGMAN RD
 STE 0944
 FORT BELVOIR VA 22060-6218

1 DIRECTOR
 US ARMY RESEARCH LAB
 IMNE ALC HRR
 2800 POWDER MILL RD
 ADELPHI MD 20783-1197

1 DIRECTOR
 US ARMY RESEARCH LAB
 RDRL CIM L
 2800 POWDER MILL RD
 ADELPHI MD 20783-1197

1 DIRECTOR
 US ARMY RESEARCH LAB
 RDRL CIM P
 2800 POWDER MILL RD
 ADELPHI MD 20783-1197

1 DIRECTOR
 US ARMY RESEARCH LAB
 RDRL D
 2800 POWDER MILL RD
 ADELPHI MD 20783-1197

ABERDEEN PROVING GROUND

1 DIR USARL
 RDRL CIM G (BLDG 4600)